REMARKS/ARGUMENTS

Claims 1-26 are pending in this application. Claims 1-26 stand rejected. No claims have been amended, added, or canceled in this paper.

Claim Rejections - 35 USC § 102

Claims 1-3, 6-11, 14-19, and 22-26 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Garthwaite (U.S. Pub. No. 2002/0161792 A1). As to claims 1, 9, 17, and 25-26, the Office Action asserts that Garthwaite (at p. 6, ¶ 63 and p. 7, ¶ 90 – p. 8, ¶ 93) discloses all of the claimed elements.

Applicant traverses the rejection and respectfully requests that it be withdrawn. Garthwaite does not disclose "an object allocation routine which stores *an object of a particular type* in one of a plurality of logical partitions in the heap dependent on a *predefined category assigned to the object type*" (emphasis added), as required by claim 1 as currently presented. As described in the specification of the present application, an example of an "object type" in the Java™ programming language is the char[] object type, which is used to store character strings (p. 8, line 23 – p. 9, line

5). Other well-known examples of object types include object types for storing integers and those for storing floating-point numbers.

The specification of the present application further provides an example of assigning predefined categories to different object types. For example, in one embodiment there are two spaces in which objects may be stored: hot space and cold space. Correspondingly, there are two object categories: hot and cold. Some object types may have a preference for being stored in hot space, while other object types may have a preference for being stored in cold space. The preferred space of an object type indicates the category of the object type. For example, the hot category is assigned to object types that are preferred to be stored in hot space, while the cold category is assigned to object types that are preferred to be stored in cold space. (*See* p. 7, line 12 - p. 9, line 5.)

Garthwaite does not disclose any assignment of predefined categories to object types. Garthwaite does not, therefore, disclose "stor[ing] an object of a particular type in one of a plurality of logical partitions in the heap dependent on a predefined category assigned to the object type," as required by claim 1. Rather, Garthwaite discloses placing "popular" objects in different "cars," where an object is defined as "popular" based on *the number of other objects that reference it* (see Garthwaite at p. 8, ¶ 94). The distinction between this technique of Garthwaite and

the above-cited limitation of claim 1 is evidenced by the fact that Garthwaite does not make any use of object type in the storage of objects.

For at least this reason, claim 1 of the present application patentably distinguishes over Garthwaite. The same reasoning applies to claims 9, 17, and 25-26 of the present application. Claim 9, for example, recites "means for storing an object of a particular *type* in one of a plurality of logical partitions in the heap dependent on a predefined category assigned to the object *type*" (emphasis added). Claim 17 recites a step of "storing an object of a particular *type* in one of a plurality of logical partitions in the heap dependent on a predefined category assigned to the object *type*" (emphasis added). Claim 25 recites a collector "storing an object of a particular *type* in one of a plurality of logical partitions in the heap dependent on a predefined category assigned to the *type*" (emphasis added). Finally, claim 26 recites program code which "stores an object of a particular *type* in one of a plurality of logical partitions in the heap dependent on a predefined category assigned to the *type*" (emphasis added). Claims 9, 17, and 25-26 therefore patentably distinguish over Garthwaite for at least the same reasons as claim 1.

Claims 2-3, 6-8, 10-11, 14-16, 18-19, and 22-24 depend, either directly or indirectly, from corresponding ones of the independent

claims referenced above and therefore patentably distinguish over Garthwaite for at least the same reason.

Claim 6, which depends indirectly from claim 1, was called out specifically by the Office Action for an additional reason. The Office Action asserted that claim 6, which states that the object category is dependent on "object type mortality," reads on generational garbage collection technology generally. This is not correct. As described in Applicant's previous arguments, generational garbage collection involves performing garbage collection based on the *age* of individual objects. The term "generational garbage collection" refers to the fact that an object's age is measured by the number of *generations* the object has been in existence.

Object type mortality, in contrast, is not a measure of the *age* of an object. The mortality of a particular object type represents a measure of the probability that objects of that type will survive a future garbage collection (p. 16, lines 12-15). The mortality of an object type does not have any necessary relationship to the age of objects of that type; an old object may be of a type that has either a high or a low mortality, as may a young object.

Object types with high mortality may be assigned to the "hot" category, while object types with low mortality may be assigned to the "cold" category. Object mortality may, for example, be

estimated based on the difference between the number of bytes of the object type stored in the heap before a garbage collection and the number of bytes of the object type stored in the heap after the garbage collection (p. 4, lines 3-7).

Object mortality, therefore, is not a measure of the age of an object, which is precisely the measure used by generational garbage collection schemes. Claim 6, therefore, which recites that the object category is dependent on object type mortality, does not read generally on generational garbage collection technology.

Claim Rejections - 35 U.S.C. § 103

Claims 4, 5, 12, 13, 20, and 21 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Garthwaite (U.S. Pub. No. 2002/0161792 A1) in view of Ebrahim et al. (U.S. Pat. No. 5,930,807). Neither Garthwaite nor Ebrahim, however, either singly or in combination, teach or suggest the above-cited limitation: "an object allocation routine which stores *an object of a particular type* in one of a plurality of logical partitions in the heap dependent on a *predefined category assigned to the object type*" (emphasis added). Claims 4, 5, 12, 13, 20, and 21 include this claim by incorporation from the independent claims described above, and therefore patentably distinguish over the combination of Garthwaite and Ebrahim for at least the reasons described above.

CONCLUSIONS

Any dependent claims not specifically referenced above incorporate the limitations of the independent claims from which they depend, and therefore are patentable for at least the same reasons.

If the Examiner considers the arguments presented herein not to be persuasive, the Applicant respectfully requests that the Examiner contact the Applicant to schedule an interview at a mutually convenient time.
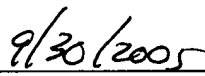
If this response is not considered timely filed and if a request for extension of time is otherwise absent, applicant hereby requests any extension of time. Please charge any fees or make any credits, to Deposit Account No. 08-2025.

Respectfully submitted,

_____   ___9/30/2005___
Robert Plotkin, Esq.   ·   Date
Reg. No. 43,861

Robert Plotkin, P.C.
91 Main Street, Suite 204
Concord, MA 01742-2527
Tel: (978) 318-9914
Fax: (978) 318-9060